



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/709,917

06/04/2004

Evan P. Ireland

SYB/0101.01

3916

31779

7590

09/12/2008

JOHN A. SMART

201 LOS GATOS

SARATOGA RD, #161

LOS GATOS, CA 95030-5308

EXAMINER

WANG, RONGFA PHILIP

ART UNIT

PAPER NUMBER

2191

MAIL DATE

DELIVERY MODE

09/12/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/709,917	Applicant(s) IRELAND, EVAN P.	
	Examiner PHILIP WANG	Art Unit 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 30 June 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-61 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-61 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

Detail Action

1. This office action is in response to the amendment filed on 6/30/2008.
2. Per Applicant's request, claims 1, 5, 24-26, 47, 60, and 61 have been amended.
3. Claims 1-61 are pending.

Priority

4. The priority date considered for this application is 3/29/2004.

Specification

5. The disclosure is objected to because of the following informalities:

The disclosure is objected to because it contains an embedded hyperlink and/or other form of browser-executable code. Applicant is required to delete the embedded hyperlink and/or other form of browser-executable code. See MPEP § 608.01. For example, [0034], or [0036] contains a hyperlink. For example, amending "www.eclipse.org" to "eclipse.org" does not change the fact that "eclipse.org" still refers to an URL link on the Internet.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. The 35 U.S.C. 101 rejections of claims 26-46 have been withdrawn in view of the Applicant's amendment to the claims.

Claim Objections

7. Objections to Claims 24, 25, 60 and 61 have been withdrawn in view of the Applicant's amendment to the claims.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claims 1-46 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

For example, claims 1 and 26 include the limitation of "a static field of type Component". It appears "Component" appears to be proper name of certain type. However, the claim language does not include any description of what "type Component" is. Dependent claims of the above independent claims suffer the same deficiency.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Art Unit: 2191

9. Claims 1-5, 15-19, 21-26, 28-30, 33, 40-43, and 45-46 are rejected under 35 U.S.C. 102(b) as being unpatentable over DeGroot et al. (US Patent No. 6,182,277) in view of McGurrin et al. (USPN 5,913,063).

As per claim 1,

DeGroot et al. disclose

- adding a static field of type Component to a program class of the program to create a component (c4: 40-43, "...method of an object..." A method is component in a class and is therefore added.);
- defining at least one attribute specifying declaratively behavior to be added to the program, wherein said at least one attribute comprises active metadata used to generate program code for inclusion in the program (c4: 40-53, "...declarative techniques...that define object behavior...");
- associating said at least one attribute with the component (c4:1-5, "...associate the declarative statements to ...on the object..."); and
- in response to instantiation of the component at runtime, (c2: 24-25, "...when the method on that object is called..." calling the method instantiates the object.)

DeGroot et al. do not specifically disclose

Art Unit: 2191

- generating a subclass based on the program class and said at least one attribute, the subclass including dynamically generated program code based on said at least one attribute;

However, McGurrin et al. disclose

- generating a subclass based on the program class and said at least one attribute, the subclass including dynamically generated program code based on said at least one attribute (c2:27-35, "Some visual coding tools generate source code for object oriented programming languages. With these visual coding tools, a generic object class is typically defined for every type of visual element. When a user draws a particular visual element, the visual coding tool generates source code for (1) a new object class that inherits from the corresponding generic object class, where the attributes of the subclass are initially set to those reflected in the element drawn by a user, and (2) an instance of the new object class. " where user entered attribute and subclass that are generated is described.)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of McGurrin into the teachings of DeGroot to include the limitation discloses by McGurrin. The modification would be obvious to one of ordinary skill in the art to want to simplifying the process of creating object subclasses based upon existing object classes as suggested by McGurrin et al. (c5: 6-10).

Art Unit: 2191

As per claim 2,

the rejection of claim 1 is incorporated,

DeGroot et al. disclose

- wherein said defining step includes defining a particular attribute using active metadata, so as to provide a mechanism for generation of program code from said particular attribute(c2:26-29, "...access to metadata...").

As per claim 3,

the rejection of claim 2 is incorporated,

DeGroot et al. disclose

- wherein said active metadata dynamically generates code for inclusion in a subclass based on the program class(c2:26-29).

As per claim 4,

the rejection of claim 1 is incorporated,

DeGroot et al. disclose

Art Unit: 2191

- wherein the generating step includes generating a subclass comprising an instance of a declared component class(c2:15-18, "...new subclass...").

As per claim 5,

the rejection of claim 1 is incorporated,

DeGroot et al. disclose

- Wherein the generating step includes generating a subclass comprising an instance of component class declared as abstract (c4:20-25, "...An abstract specification...").

As per claim 15,

the rejection of claim 1 is incorporated,

DeGroot et al. disclose

- wherein said defining step includes defining attributes for a superclass from which the program class inherits(c1:64-66, "..inheritance...passing attributes...").

As per claim 16,

the rejection of claim 1 is incorporated,

DeGroot et al. disclose

Art Unit: 2191

- wherein said defining step includes defining attributes for the program class' package from which the program class inherits(c7: 44-45, "...package...").

As per claim 17,

the rejection of claim 1 is incorporated,

DeGroot et al. disclose

- wherein said defining step includes defining attributes for an interface from which the program class inherits(c2:21-23, "...object interface...").

As per claim 18,

the rejection of claim 17 is incorporated,

DeGroot et al. disclose

- wherein said generating step includes generating an instance of a subclass to mock the behavior of the interface(c2:30-34, "...augment or change...object...").

As per claim 19,

the rejection of claim 1 is incorporated,

DeGroot et al. disclose

Art Unit: 2191

- wherein said generating step includes generating code for a non—abstract method body based on an attribute defined for an abstract method(c2:21-25, “...subclassing...pointing to a new function...”).

As per claim 21,

the rejection of claim 1 is incorporated,

DeGroot et al. disclose

- adding expected calls as instances of anonymous inner classes of the program;
and
applying runtime introspection by a generated subclass to verify a sequence of expected calls(c12:32-41, “...inner invocation of the actual method...”).

As per claim 22,

the rejection of claim 1 is incorporated,

DeGroot et al. disclose

- wherein the component registers itself with a repository when the component is initially activated(c8:5-10, “...object register...”).

As per claim 23,

the rejection of claim 22 is incorporated,

Art Unit: 2191

DeGroot et al. disclose

- wherein the repository can be queried to determine components that are active(c3:11-15, "...the user may query...objects...").

As per claim 24,

DeGroot et al. disclose

- A computer—readable medium having processor-executable instructions for performing the method of claim 1 (see rejection of claim 1).

As per claim 25,

DeGroot et al. disclose

- A downloadable set of processor-executable instructions for performing the method of claim 1(see rejection of claim 1)..

As per claim 26,

DeGroot et al. disclose

- (see rejection of claim 1).

Art Unit: 2191

As per claim 28,

DeGroot et al. disclose

- the rejection of claim 26 is incorporated, wherein the subclass is a subclass of an abstract class(c4:20-25, "...An abstract specification...").

.

As per claim 29,

the rejection of claim 26 is incorporated,

DeGroot et al. disclose

- wherein said at least one declarative attribute includes active metadata, so as to provide a mechanism for generation of program code(c2:26-29, "...access to metadata...").

As per claim 30,

the rejection of claim 29 is incorporated,

DeGroot et al. disclose

- wherein said active meta— data dynamically generates code for inclusion in the subclass of the program class (c2: 14-25, "...subclassing technique

Art Unit: 2191

permit...to generate a new method on a subclass..";c8: 6-8,
"...may be added at run time..."; c2:40-42, "...the subclassing
technique requires re-compiling the code...")..

As per claim 33,

the rejection of claim 32 is incorporated,

DeGroot et al. disclose

- wherein the module for generating loads the class containing static attributes
before subclass generation(c2: 14-25, "...subclassing technique
permit...to generate a new method on a subclass..";c8: 6-8,
"...may be added at run time..."; c2:40-42, "...the subclassing
technique requires re-compiling the code...").

As per claim 40,

the rejection of claim 26 is incorporated,

- See rejection of claim 15.

As per claim 41,

the rejection of claim 26 is incorporated,

- See rejection of claim 17.

Art Unit: 2191

As per claim 42,

the rejection of claim 41 is incorporated,

- See rejection of claim 18.

As per claim 43,

the rejection of claim 26 is incorporated,

DeGroot et al. disclose

- wherein the module for generating generates code for a non—abstract system body based on an attribute defined for an abstract method(c4:13-15, “...augmentation...at instance level...”).

As per claim 45,

the rejection of claim 26 is incorporated,

- See rejection of claim 22.

As per claim 46,

the rejection of claim 45 is incorporated,

- See rejection of claim 23.

Art Unit: 2191

10. Claims 6-14, 20, 31-39, and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over DeGroot et al. (US Patent No. 6,182,277), McGurrin et al. (USPN 5,913,063) in view of Foster (US Patent No. 7103885).

As per claim 6,

the rejection of claim 1 is incorporated,

DeGroot et al./McGurrin et al. do not specifically disclose

- wherein said defining step includes defining at least one attribute based on comments in source code of the program class.

However, Foster discloses

- wherein said defining step includes defining at least one attribute based on comments in source code of the program class(c7:53-57, "...comment field...a tag relating to an attribute...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Foster into the teachings of DeGroot et al./McGurrin et al. to include the limitation discloses by Foster . The modification would be obvious to one of ordinary skill in the art to want to process software module based attributes in the comments as suggested by Foster (see col. 2, 2nd paragraph).

As per claim 7,

the rejection of claim 6 is incorporated,

Art Unit: 2191

DeGroot et al. disclose

- precompiling a class containing static attributes from said comments(c2:40-42, "...recompiling...").

As per claim 8,

the rejection of claim 7 is incorporated,

DeGroot et al. disclose

- loading the class containing static attributes before subclass generation when a component is instantiated(c2:40-43, "...reloading both the old and new code.").

As per claim 9,

the rejection of claim 6 is incorporated,

DeGroot et al. disclose

- defining an automated mapping between attribute syntax in comments and attribute syntax as expressed in generated program code(c3:40-45, "...maps...").

As per claim 10,

the rejection of claim 1 is incorporated,

Art Unit: 2191

Foster discloses

- Said step defining step includes defining at least one attribute in a property file external to the program class(c5:35-38, "...management file...attribute...").

As per claim 11,

the rejection of claim 10 is incorporated,

Foster discloses

- compiling a class containing dynamic attributes from said property file(c5:35-38, "...management file...attribute...").

As per claim 12,

the rejection of claim 11 is incorporated,

DeGroot et al. disclose

- loading the class containing dynamic attributes before subclass generation when a component is instantiated(c2:40-42, "...recompiling...").

As per claim 13,

the rejection of claim 10 is incorporated,

DeGroot et al. disclose

Art Unit: 2191

- defining an automated mapping between attribute syntax in a property file and attribute syntax as expressed in generated program code(c3:40-45, "...maps...").

As per claim 14,

the rejection of claim 10 is incorporated,

DeGroot et al. disclose

- wherein attributes in the property file comprise property name and property value pairs(c6:35-38, "...values of parameters...").

As per claim 20,

the rejection of claim 1 is incorporated,

Foster discloses

- wherein said generating step includes generating program code based on comments in a source file(c7:53-57, "...comment field...a tag relating to an attribute...").

As per claim 31,

the rejection of claim 26 is incorporated,

Art Unit: 2191

- wherein the attribute module provides for defining at least one attribute based on comments in source code of the program class(see rejection of claim 6).

As per claim 32,

the rejection of claim 31 is incorporated,

a precompiler for precompiling a class containing static attributes from said comments().

As per claim 34,

the rejection of claim 31 is incorporated,

- see rejection of claim 13.

As per claim 35,

the rejection of claim 26 is incorporated,

- See rejection of claim 10.

As per claim 36,

the rejection of claim 35 is incorporated,

- see rejection of claim 12.

Art Unit: 2191

As per claim 37,

the rejection of claim 36 is incorporated,

DeGroot et al. disclose

- wherein the module for generating loads the class containing dynamic attributes before subclass generation when a component is instantiated (c2: 14-25, "...subclassing technique permit...to generate a new method on a subclass.."; c8: 6-8, "...may be added at run time..."; c2:40-42, "...the subclassing technique requires re-compiling the code...").

As per claim 38,

the rejection of claim 35 is incorporated,

DeGroot et al. disclose

- an automated mapping between attribute syntax in a property file and attribute syntax as expressed in generated program code(c3:40-45, "...maps...").

As per claim 39,

the rejection of claim 35 is incorporated

- See rejection of claim 14.

Art Unit: 2191

As per claim 44,

the rejection of claim 26 is incorporated,

- See rejection of claim 20.

11. Claims 47-61 are rejected under 35 U.S.C. 103(a) as being unpatentable over DeGroot et al. (US Patent No. 6,182,277) in view of Zhang et al. (US Patent No. 2003/0055936).

As per claim 47,

DeGroot et al. disclose

- defining at least one attribute specifying declaratively behavior which is desired to be added to an application without access to the application source code (c4: 40-53, "...declarative techniques...that define object behavior...");
- wherein said at least one attribute comprises active metadata used to generate code adding behavior to the application (c4: 40-53, "...declarative techniques...that define object behavior...");
- adding behavior to the application based on said at least one attribute (c2: 14-25, "...subclassing technique permit...to generate a new method on a subclass.."; c8: 6-8, "...may be added at run time...");

Art Unit: 2191

c2:40-42, "...the subclassing technique requires re-compiling the code...").

DeGroot et al. do not specifically disclose

- and generating a subclass which includes dynamically generated code,

However, McGurrin et al. disclose

- and generating a subclass which includes dynamically generated code,(c2:27-35, "Some visual coding tools generate source code for object oriented programming languages. With these visual coding tools, a generic object class is typically defined for every type of visual element. When a user draws a particular visual element, the visual coding tool generates source code for (1) a new object class that inherits from the corresponding generic object class, where the attributes of the subclass are initially set to those reflected in the element drawn by a user, and (2) an instance of the new object class. ", where user entered attribute and subclass that are generated is described.)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of McGurrin into the teachings of DeGroot to include the limitation discloses by McGurrin. The modification would be obvious to one of ordinary skill in the art to want to simplifying the process of creating object subclasses based upon existing object classes as suggested by McGurrin et al. (c5: 6-10).

Art Unit: 2191

DeGroot et al..McGurrin et al. do not specifically disclose

- storing said at least one attribute in a properties file external to the application;
creating a dynamic attributes class based on the properties;
compiling the application and the dynamic attributes class;

However Zhang et al. disclose

- storing said at least one attribute in a properties file external to the application;
creating a dynamic attributes class based on the properties file ([0075], "...a text file for...each dynamic attribute class...");
compiling the application and the dynamic attributes class; ([0072], "...a dynamic attribute class...compile the dynamic attribute class...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Zhang et al. into the teachings of DeGroot et al..McGurrin et al. to include the limitation discloses by Zhang et al. . The modification would be obvious to one of ordinary skill in the art to want to be able to define new dynamic attributes as suggested by Zhang et al. ([0013]).

As per claim 48,

the rejection of claim 47 is incorporated,

- DeGroot et al. disclose
wherein said defining step includes defining a particular attribute using active metadata, so as to provide a mechanism for generation of program code from

Art Unit: 2191

said particular attribute(c2:26-29, "...access to metadata...").

As per claim 49,

the rejection of claim 48 is incorporated,

- see rejection of claim 3.

As per claim 50,

the rejection of claim 47 is incorporated,

DeGroot et al. disclose

- wherein said defining step includes defining an attribute for overriding a method of the application(c2:50, "...method overriding...").

As per claim 51,

the rejection of claim 47 is incorporated,

DeGroot et al. disclose

- wherein said defining step includes defining an attribute for extending a method of the application(c4:26-30, "...extended definition...").

Art Unit: 2191

As per claim 52,

the rejection of claim 47 is incorporated,

DeGroot et al. disclose

- loading the class containing dynamic attributes before generating the subclass(c12:36-40, "...dynamically loaded...").

As per claim 53,

the rejection of claim 47 is incorporated,

DeGroot et al. disclose

- defining an automated mapping between attribute syntax in the properties file and attribute syntax as expressed in generated program code(c3:40-45, "...maps...").

As per claim 54,

the rejection of claim 47 is incorporated,

DeGroot et al.

- attribute in the properties file comprise property name and property value pairs(c6:35-38, "...values of parameters...").

As per claim 55,

the rejection of claim 47 is incorporated,

DeGroot et al. disclose

Art Unit: 2191

- wherein said creating step includes creating a dynamic attributes class using a pre— compiler(c2:40-42, “...recompiling...”).

As per claim 56,

the rejection of claim 47 is incorporated,

Zhang et al. disclose

- wherein said creating step includes creating a dynamic attributes class at runtime([0070], “...creating of dynamic attribute classes...”).

As per claim 57,

the rejection of claim 47 is incorporated,

Zhang et al. disclose

- wherein said compiling step includes using a Java compiler (JAVAC)([0030], “...Java classes are compiled...”).

As per claim 58,

the rejection of claim 47 is incorporated,

DeGroot et al. disclose

Art Unit: 2191

- wherein said generating step includes using a precompiler(c2:40-42, "...recompiling...").

As per claim 59,

the rejection of claim 47 is incorporated,

Zhang et al.

- wherein said generating step includes using a runtime compiler(c2:40-42, "...recompiling...").

. As per claim 60,

- see rejection of claim 47.

As per claim 61,

- see rejection of claim 47.

As per claim 27,

the rejection of claim 26 is incorporated,

DeGroot et al. do not specifically disclose

- wherein the subclass adds tracing behavior to a program.

Art Unit: 2191

However Zhang et al. disclose

- wherein the subclass adds tracing behavior to a program([0051], "...debug tracing...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Zhang et al. into the teachings of DeGroot et al. to include the limitation discloses by Zhang et al. . The modification would be obvious to one of ordinary skill in the art to want to be able to define new dynamic attributes as suggested by Zhang et al. ([0013]).

Response to Arguments

12. Applicant's arguments with respect to claims have been considered but are moot in view of the new ground(s) of rejection. Additionally the Applicant argues that Foster's attributes is dramatically different from the applicant's invention. The claim language only shows the attribute may comprises active metadata. Based on what is being presented, attribute in information that can be based upon to generate code. The claim language does not appear to include elements that sufficiently enough to support the differentiation the Applicant has argued. It is recommended future amendments will include clear definition of terms used in the claim language and also include specific elements or steps or how and what is included in the invention.

Conclusion

Art Unit: 2191

THIS ACTION IS MADE FINAL. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

It is noted that any citation [[s]] to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. [[See, MPEP 2123]]

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Philip Wang whose telephone number is 571-272-5934. The examiner can normally be reached on Mon - Fri 8:00AM - 4:00PM. Any inquiry of general nature or relating to the status of this application should be directed to the TC2100 Group receptionist: 571-272-2100.

Art Unit: 2191

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Wei Y Zhen/

Supervisory Patent Examiner, Art Unit 2191